HIGH-PERFORMANCE PARALLEL INTERFACE -

6400 Mbit/s Physical Switch Control

(HIPPI-6400-SC)

November 4, 1996

Secretariat:

Information Technology Industry Council (ITI)

ABSTRACT: This revision of HIPPI-6400-SC provides a protocol for controlling physical layer switches which are based on the High-Performance Parallel Interface at 6400 Mbits/s (HIPPI-6400-PH), a simple high-performance point-to-point interface for transmitting digital data at peak data rates of 6400 Mbit/s between data-processing equipment.

NOTE:

This is an internal working document of X3T11, a Technical Committee of Accredited Standards Committee X3. As such, this is not a completed standard. The contents are actively being modified by X3T11. This document is made available for review and comment only. For current information on the status of this document contact the individuals shown below:

POINTS OF CONTACT:

Roger Cummings (X3T11 Chairman)
Distributed Processing Technology
140 Candace Drive
Maitland, Fl 32751
(407) 830-5522 x348, Fax:(407) 260-5366
E-mail: cummings_roger@dpt.com

Carl Zeitler (X3T11 Vice-Chairman)
IBM Corporation, MS 9440
11400 Burnet Road
Austin, TX 78758
(512) 838-1797, Fax: (512) 838-3822
E-mail: zeitler@ausvm6.vnet.ibm.com

Roger Ronald (HIPPI-6400-SC Technical Editor)
E-Systems
MS 35300 HD
PO Box 660023
Dallas, TX 75266-0023
(214) 205-8043, Fax: (214) 272-8144
E-mail: rronald@esy.com

Comments on Rev 0.20

This is a preliminary document. The first draft (rev 0.01) was presented and reviewed for the first time in March 1996. The second revision (rev .10) was reviewed on May 9th and 10th in Dallas. This revision corrects errors discovered at that time and continues the process of documentation.

Rev bars are now included in this revision of the document except for cases of minor punctuation or spelling error correction.

Major changes from the previous revision include:

- new definitions added for alternate pathing, final destination and original source
- changing the requirements for selection of the switch port to allow alternate pathing
- beefing up the paragraph on error checking required in fabric
- adding information on micropacket interleaving
- clarifying the paragraph on congestion management
- adding reserved addresses from the HIPPI-800-SC standard
- adding more information on routing to the informative appendix

In addition to discussing these changes, it is expected that the next meeting (Santa Fe, June 10-11) will cover usage of administrative packets.

Comments on Rev 0.30

This revision was started to collect changes and additions made during and after the June ANSI meeting held in Santa Fe, NM on June 10th thru the 12th.

Major changes from the previous revision include:

- Added definitions
- General clean-up
- Pruning of sections detailing alternative addressing approaches (alternate pathing, broadcast, and multicast)
- Moved congestion management paragraph to reside within the general section on error protection
- Removed requirement of 4 micropacket message support for in-band communications
- Split switching, bridging, and routing into three appendices while adding text and examples

Comments on Rev 0.40

This revision was started to collect changes and additions made during and after the July HIPPI-6400 working group meeting held in San Jose on July 11th and 12th.

Major changes from the previous revision include:

- Definitions added for administrator, fabric, log, and switch
- Switch addressing references to optional modes reduced to a minimum
- Address restrictions for inter-operation with HIPPI-800 removed from section 6
- Removed e-mail list instructions from this page for obsolete mail groups

Comments on Rev 0.45

This revision was started to collect changes and additions made during and after the August HIPPI-6400 ANSI meeting held in Honolulu on August 5-7, 1996. Because the document has not been reviewed line-by-line since the July working group meeting, change bars still include the revision 0.4 changes.

Major changes from the previous revision include:

- Updated definitions and acronyms to follow the lead of HIPPI-6400-PH
- Removal of comments that this specification would describe switch-to-switch negotiation of address configuration.
- Information and procedures for using admin micropackets for topology discovery.
- Information and procedures for using admin micropackets for logical address assignment.
- Replacement of 16 bit logical addresses with 48 bit Universal LAN Addresses (ULAs) and provision for optional operation using 16 of the 48 bits.
- Removed requirement to support 64K switch addresses.
- Changed the limit for the maximum count of micropackets that may be sent on a single VC before interleaving traffic from other VCs from 65 to 66 (to match the limit for a VC0 message).
- Updated text to reflect decision that all micropackets except Header micropackets (not just Data micropackets) will be treated as part of a message following a Header micropacket.

Comments on Rev 0.50

This revision was started to collect changes and additions made during and after the September and October HIPPI-6400 ANSI meetings. Because the document has not been reviewed line-by-line since the July working group meeting, change bars still include the revision 0.4 changes.

There are no major changes from the previous revision.

Please help us in this development process by sending comments, corrections, and suggestions to the Technical Editor, Roger Ronald @ E-Systems via e-mail (rronald@esy.com).

Table Of Contents

List Of Figures

List Of Tables

Foreword (This Foreword is not part of American National Standard X3.xxx-199x.)

This American National Standard specifies the behavior and control for HIPPI-6400 physical layer switches. HIPPI-6400 is an efficient high-performance point-to-point interface. HIPPI-6400 physical layer switches may be used to give the equivalent of multi-drop capability, connecting together multiple data processing equipments.

This standard provides an upward growth path for legacy HIPPI-based systems.

This document includes annexes which are informative and are not considered part of the standard.

Requests for interpretation, suggestions for improvement or addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

> (List of X3 Committee members to be included in the published standard by the ANSI Editor.)

Subcommittee X3T11 on Device Level Interfaces, which developed this standard, had the following participants:

> (List of X3T11 Committee members, and other active participants, at the time the document is forwarded for public review, will be included by the Technical Editor.)

Introduction

This 6400 Mbits/second High-Performance Parallel Interface, Physical Switch Control (HIPPI-6400-SC) standard defines the control for HIPPI-6400 physical layer switches. HIPPI-6400 is an efficient high-performance point-to-point interface. Small fixed-size micropackets provide an efficient, low-latency, structure for small messages, and a building block for large messages. HIPPI-6400 physical layer switches may be used to give the equivalent of multi-drop capability, connecting together multiple data processing equipments.

Characteristics of this HIPPI-6400 physical switch control protocol include:

- Support for 48 bit Universal LAN Addresses (ULAs)
- Support for restricted mode operation with a 16 bit subset of the ULA
- Procedures for use of admin micropackets to automate address assignments
- Ability to span multiple physical layer switches within a fabric
- Support for physical layer switches with differing numbers of ports, all within the same fabric
- Specified reserved addresses to aid address self-discovery, switch management, and switch control
- Support for 4 Virtual Channels

American National Standard
for Information Technology –

High-Performance Parallel Interface –

6400 Mbit/s Physical Switch Control (HIPPI-6400-SC)

1.0 Scope

This American National Standard provides switch control for physical layer switches using the 6400 Mbits/second High-Performance Parallel Interface (HIPPI-6400), a high-performance point-to-point interface between data-processing equipment.

The purpose of this standard is to facilitate the development and use of the HIPPI-6400 in computer systems by providing common physical switch control. The standard provides switch control structures for physical layer switches interconnecting computers, high-performance display systems, and high-performance, intelligent block-transfer peripherals. This standard also applies to point-to-point HIPPI-6400 topologies.

Specifications are included for:

- Interleaving of Virtual Channels (VCs) within a physical channel
- Selection of Messages for transmission on physical channels
- Self discovery of configuration information

2.0 Normative references

The following American National Standard contains provisions which, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard listed below.

ANSI X3.183-1991, High-Performance Parallel Interface – Mechanical, Electrical, and Signalling Protocol Specification (HIPPI-PH).

ANSI X3.210-1992, High-Performance Parallel interface, Framing Protocol (HIPPI-FP).

ANSI X3.222-1993,High-Performance Parallel interface, Physical Switch Control (HIPPI-SC).

ANSI X3.xxx-199x, High Performance Parallel Interface 6400 Mbits/s Physical Layer (HIPPI-6400-PH)

3.0 Definitions and conventions

3.1 Definitions

For the purposes of this standard, the following definitions apply.

3.1.1 administrator: A station management entity providing external management control.

3.1.2 alternate pathing: Capability to address a Message to select from a group of ports based upon defined criteria.

3.1.3 Destination: The equipment that receives the data.

3.1.4 fabric: All of the switching equipment connected together in a configuration.

3.1.5 Final Destination: The equipment that receives, and operates on, the payload portion of the micropackets. This is typically a host computer system, but may also be a translator, bridge, or router.

3.1.6 HIPPI-PH: High-Performance Parallel Interface - Mechanical, Electrical, and Signalling Protocol Specification (HIPPI-PH), ANSI X3.183-1991. Data is transmitted in parallel over copper twisted-pair cables at 800 or 1600 Mbits per second.

3.1.7 HIPPI port: A HIPPI-6400-PH, or HIPPI-PH, Source or Destination.

3.1.8 in-band: Switch control communications accomplished over a HIPPI-6400 link. As opposed to out-of-band (using an alternative communication channel).

3.1.9 link: A full-duplex connection between HIPPI-6400-PH devices.

3.1.10 log: The act of making a record of an event for later usage.

3.1.11 logical address: An address stored in a Source or Destination address field that uniquely identifies an Originating Source or Final Destination.

3.1.12 micropacket: The basic transfer unit consisting of 32 data bytes and 64 bits of control information.

3.1.13 Message: An ordered sequence of one or more micropackets which have the same VC. The first micropacket is a Header micropacket. The last micropacket, which may also be the first micropacket, has the TAIL bit set.

3.1.14 optional: Characteristics that are not required by HIPPI-6400-SC. However, if any optional characteristic is implemented, it shall be implemented as defined in HIPPI-6400-SC.

3.1.15 Originating Source: The equipment that generates the payload portion of the micropackets. This is typically a host computer system, but may also be a translator, bridge, or router.

3.1.16 Source: The equipment that transmits the data.

3.1.17 switch: An equipment that provides connections between HIPPI-6400 links based on this standard.

3.1.18 Virtual Channel (VC): One of four logical paths within each direction of a link.

3.2 Editorial conventions

In this standard, certain terms that are proper names of signals or similar terms are printed in uppercase to avoid possible confusion with other uses of the same words (e.g., FRAME). Any lowercase uses of these words have the normal technical English meaning.

A number of conditions, sequence parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., State, Source). Any lowercase uses of these words have the normal technical English meaning.

The word shall when used in this American National standard, states a mandatory rule or requirement. The word should when used in this standard, states a recommendation.

3.2.1 Binary notation

Binary notation is used to represent relatively short fields. For example a two-bit field containing the a binary value of 10 is shown in binary format as b'10'.

3.2.2 Hexadecimal notation

Hexadecimal notation is used to represent some fields. For example a two-byte field containing a binary value of b’1100010000000011’ is shown in hexadecimal format as x'C403'.

3.2.3 Acronyms and other abbreviations

ACK	acknowledge indication
ARP	Address Resolution Protocol
CR	credit amount parameter
CRC	cyclic redundancy check
ECRC	end-to-end CRC
HIPPI	High-Performance Parallel Interface
IP	Internet Protocol
LCRC	link CRC
MAC	Media Access Control
ns	nanoseconds
RIP	Routing Information Protocol
RSEQ	receive sequence number
TSEQ	transmit sequence number
ULA	universal LAN MAC address
VC	virtual channel
VCR	virtual channel credit selector
μs	microseconds

4.0 System overview

This paragraph provides an overview of the structure, concepts, and mechanisms used in HIPPI-6400-SC.

4.1 Switch function

HIPPI-6400 switches provide a method to send Messages from a Source port to a Destination port. Each Message travels on one of the four Virtual Channels (VCs) available in HIPPI-6400-PH (see HIPPI-6400-PH for assignments of Message type to VC). All of the micropackets of a Message are transmitted on a single VC, i.e., the VC number does not change as the micropackets travel from the Originating Source to the Final Destination over one or more links.

Different VCs are interleaved on the physical channel allowing up to four Messages to proceed to a Destination or from a Source at any given time.

During transfer of a Message, the VC in use is busy and is unavailable for use by other Messages involving the same Source or Destination ports.

4.2 Micropacket

Micropackets are the basic transfer unit for HIPPI-6400. As described in HIPPI-6400-PH, a micropacket is composed of 32 data bytes and 64 bits of control information.

The 64 bits of control information in each micropacket includes parameters for physical (PH) layer functions and for switch control (SC) functions. These functions include:

- selecting a VC
- detecting missing micropackets
- denoting the types of information in the micropacket
- marking the last micropacket of a Message
- signalling that the Message was truncated at its originator, or damaged en-route, and should be discarded

Table 1 describes the information that the switch fabric carries from a Destination HIPPI-6400-PH port to a Source HIPPI-6400-PH port. Table 2 and Table 3 describe the information that a switch fabric uses to determine micropacket routing.

Table 1. Data carried through fabric

Description	Size
ERROR	1 Bit
TAIL	1 Bit
VC	2 Bits
TYPE	4 Bits
ECRC	16 Bits
Payload Data	32 Bytes

Table 2. Data used to route 1st micropacket in a Message

Description	Size
TAIL	1 Bit
VC	2 Bits
TYPE	4 Bits
Payload Data	32 Bytes

Table 3. Data Used to Route Subsequent Micropackets in a Message

Description	Size
TAIL	1 Bit
VC	2 Bits
TYPE	4 Bits

Table 4 contains information that can be used to determine whether the micropacket contains errors and a means to report discovered errors.

Table 4. Data used for error checking and reporting

Description	Size
ERROR	1 Bit
TYPE	4 Bits
ECRC	16 Bits
Payload Data	32 Bytes

Note that there is information used by the switch fabric that also is carried through it.

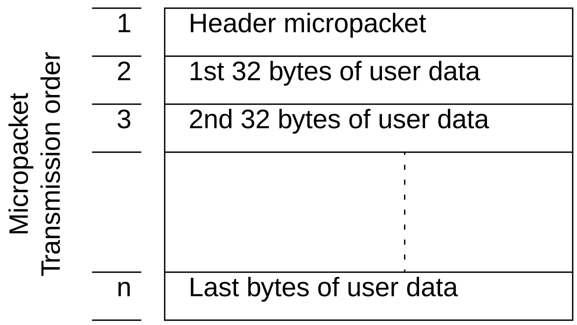

Figure 1. Message format

4.3 Message

As shown in Figure 1, Messages are logical groups of micropackets which have the same VC. The first micropacket of a Message, i.e., the Header micropacket, contains information used to route through a HIPPI-6400 fabric (see Figure 2) as well as other information as specified in HIPPI-6400-PH. The last micropacket of the Message is marked with the TAIL bit.

5.0 Switch routing

5.1 Micropacket data transferred through fabric

A HIPPI-6400 switch shall pass the information shown in Table 1 through the fabric. Micropacket data payload, the TAIL bit, the TYPE field, the VC field, and the ECRC shall not be modified while passing through a switch fabric. The ERROR bit shall be transferred as set if it was received as set. If the ERROR bit is received as not set, the bit may be set to indicate a switch detected error as described in Paragraph 5.4.

5.2 Routing of Header micropacket (first micropacket in a Message)

Figure 2 shows part of the Header micropacket. The complete specification is provided in HIPPI-6400-PH.

Within the Header micropacket, the Destination ULA specifies the address where a Message is to be sent.

The micropacket TYPE field identifies a micropacket as a Header micropacket.

TAIL = 1 on a Header micropacket indicates that there are no other micropackets beyond the Header to be sent.

The micropacket VC field specifies one of four logical paths and shall be used to address the appropriate Destination VC (micropackets traverse a fabric on a single VC and never cross VCs).

Switches shall support independent address mapping for each input port. This permits mapping the same logical address value to different output ports based upon which input port received the micropacket. See Annex A for an explanation of input port specific switching functionality.

5.2.1 Switch addressing

Switches shall support a mode of operation that provides in-order delivery of all micropackets on a VC from an Originating Source to a Final Destination.

Switches may also provide optional modes of operation including alternate pathing, broadcast, and multicast. These optional modes of operation are not covered by this standard and may not guarantee in-order Message delivery.

5.3 Routing of subsequent micropackets in a Message

Subsequent micropackets in a Message (identified by the TYPE field as any type except a Header) shall be delivered to the same fabric Destination port that the Header micropacket addressed.

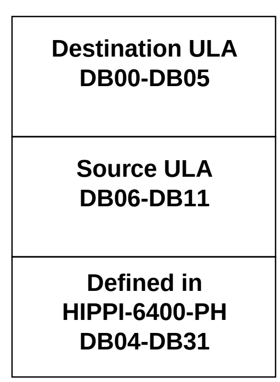

Figure 2. Header micropacket addressing

The VC field shall be used to distinguish which Message the micropacket belongs to (of the four VCs supported).

When a micropacket is received with the TAIL bit = 1, it indicates that a Message is ended with this micropacket.

5.4 Error protection

If an uncorrectable error is detected in a micropacket that is forwarded, the switch shall set the ERROR bit for that micropacket.

Detected errors shall be logged or counted.

5.4.1 Mandatory error checking

The switch fabric shall pass the unchanged ECRC with each micropacket as described in HIPPI-6400-PH.

Before sending any micropacket over a HIPPI-6400 link, the switch shall validate the ECRC and set the ERROR bit if the ECRC indicates an error as specified in HIPPI-6400-PH.

5.4.2 Optional error checking

The switch fabric may verify the validity of the ECRC at any point within the fabric.

The switch may also provide additional error detection or correction for internal data errors.

5.4.3 Congestion management

Time-out mechanisms defined in HIPPI-6400-PH will act to prevent switch congestion due to lack of progress on a HIPPI-6400 link, so long as the Source end of the link is functional. However, failures in switch Source ports can prevent this mechanism from functioning.

Switches shall protect against this failure mode by checking Source output ports for continued proper function and by discarding data destined for all failed Source output ports.

5.5 Data interleaving

There are two separate requirements for switch fairness to resolve contention for shared resources. Both micropackets and Messages shall be interleaved as described. These two interleaving processes shall be considered independent and applied without regard to one other.

5.5.1 Micropacket interleaving

Micropacket interleaving between the four VCs shall be applied on a micropacket count basis.

When a switch port has more than one VC with data available for output, the switch shall ensure that micropackets from each VC are afforded an equal opportunity for progress on a physical link.

The algorithm for choosing a micropacket from the available VCs shall allow interleaving on a frequent basis. The recommended algorithm is to interleave VC streams on a single micropacket basis.

Implementations trying to keep short Messages intact (to minimize latency) may use algorithms that interleave on other than a single micropacket basis. No implementations shall permit more than 66 micropackets from a particular VC to be transferred before moving on to the next VC. This limit allows transfer of the maximum permitted VC0 Message (2K bytes plus two header micropackets) in its entirety.

Figure 3 shows a simplified switch configuration with two input ports and one output port. Assuming that traffic is available to send to port “C” on more than one VC, a compliant switch alternates between providing output across all busy VCs on link “C”, not exceeding the limit of 66 micropackets before switching from one VC to the next VC.

5.5.2 Message interleaving

Message interleaving shall be applied whenever a current Message to an output port is completed.

When a switch has more than one input port with Messages ready for transfer to the same output port (on the same VC), the switch shall ensure that Messages from the input ports are afforded an equal opportunity for progress. All ports with pending Messages shall be serviced prior to any other port being serviced twice.

In Figure 3, an example would be if both port “A” and port “B” have multiple Messages available on their VC0 links ready to send to port “C”. In this example, Messages transferred out VC0 of port “C” are required to alternate between Messages from "A" and "B".

6.0 Addressing restrictions and reserved addresses

Although HIPPI-6400 standards provide for a 48-bit address space, the total address space is not available for all uses. Part of the range of addresses is reserved to designate the addresses of network services whose location in the network may vary, and for other network management functions. All other addresses are available for assignment to specific Destinations. The addresses assigned or reserved at the time that this standard was approved are shown in Table 5 in hexadecimal format.

Note: Later registrations will be added as an addendum to this standard, or as a revision of the standard.

Table 5. Reserved logical addresses

Start of Range	End of Range	Description
0F90	0FBF	Reserved to preserve compatibility with HIPPI-SC address trial self-discovery process
0FC0	0FDF	Reserved for local use
0FE0	0FE0	Messages pertaining to switch configuration, including HIPPI-LE Address Resolution requests as described in IETF RFC 1374 "IP and ARP on HIPPI" [1]
0FE1	0FE1	All IP protocol traffic conventionally directed to the IEEE 802.1 broadcast address as described in IETF RFC 1042 "Standard for IP transmission over 802.1 networks [2]

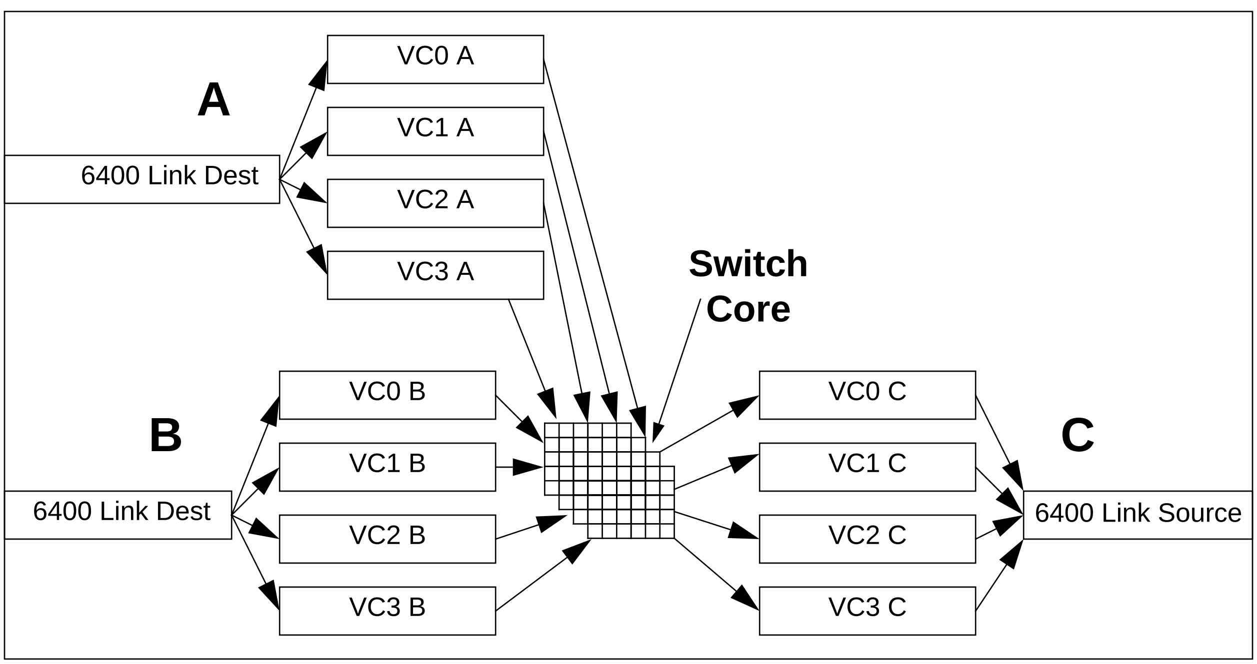

Figure 3. HIPPI-6400 Switch

Table 5. Reserved logical addresses

Start of Range	End of Range	Description
0FE2	0FE2	RFC 1112 Host extensions for IP multicasting class D addresses not assigned below [3]
0FE3	0FE3	RFC 1131 OSPF specification All Routers (Class D address 224.0.0.5) [4]
0FE5	0FE7	Reserved
0FE8	0FE8	ISO/IEC 9542:1988 CLNP ES-IS all ES's [5]
0FE9	0FE9	ISO/IEC 9542:1988 CLNP ES-IS all ES's [5]
0FEA	0FEA	ISO/IEC 10589:1992 IS-IS all level 1 IS's [6]
0FEB	0FEB	ISO/IEC 10589:1992 IS-IS all level 2 IS's [6]
0FEC	0FEC	IEEE 802.1d MAC bridging flooding
0FED	0FED	IEEE 802.1d MAC bridging Spanning Tree Protocol
0FEE	0FEE	Embedded switch management agent
0FEF	0FFC	Reserved
0FFD	0FFD	Loopback logical address for switches to use when probing other switches
0FFE	0FFE	loopback logical address for hosts to use when probing switches for the host's logical address.
0FFF	0FFF	Unknown or unassigned address. This value should never be used to address a Destination or Destinations. It can be used to indicate that the Source is unaware of its Source address or to signify an unknown logical address in higher layer protocols.

The protocols used to access these services and the means whereby these services keep track of their configuration of the network are outside the scope of this standard.

7.0 Address Configuration

In addition to switching HIPPI-6400 Messages between ports, HIPPI-6400 switches shall support in-band communications for switch management functions.

To support topology discovery and address configuration, switches shall be capable of receiving and processing micropackets with TYPE = Admin over any connected HIPPI-6400 link.

To support topology discovery and address configuration, switches shall be capable of sending micropackets of TYPE = Admin over any connected HIPPI-6400 link.

7.1 Determination of Topology

As a step in the procedure to establish a ULA for self identification (used in the Source ULA field), endpoints shall identify if they are connected to another endpoint or to a switch.

Intervening link support hardware and interface components may be present on either side of a HIPPI-6400 link. These intermediate elements will typically not contain information useful for logical address assignment. The endpoint discovering topology information shall identify these intermediate points to discover the location of an element capable of exchanging information about address configuration.

Information about the type of connected element is collected by sending an Exchange Type admin micropacket. The endpoint may directly address a destination if the appropriate admin element address information is already known, or it may use hop-count addressing to discover what is connected and how far away (in hops) the element of interest is located.

If an element responds that it is of Type Link or of Type Unknown, the probing system shall continue to the next element. Once a connected element is identified as an endpoint or switch, topology determination is complete.

In Figure 4, an example of an endpoint to endpoint link is shown. In this example, System A needs to determine the device type of System B, for address configuration. System B also needs to determine the device type of System A, for the same reason.

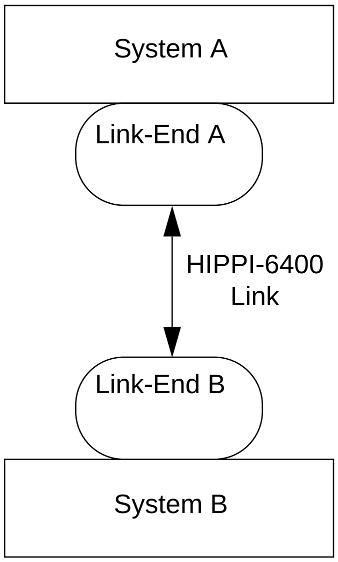

Figure 4. Endpoint to endpoint connect

The following example traces the operation of System A.

System A begins by probing each device that supports admin micropackets until it reaches the endpoint of System B.

- System A sends an Exchange Type admin micropacket to the closest point with an element address of 0xFFFFFFFF and a hop-count of 1. This will be received and processed by Link-End A. Link-End A will respond in the Return Type admin micropacket that it is a link. System A must therefore go further to reach another endpoint or switch.
- System A sends an Exchange Type admin micropacket to the next closest point with an element address of 0xFFFFFFFF and a hop-count of 2. This will be received and processed by Link-End B. Link-End B will respond in the Return Type admin micropacket that it is a link. System A must therefore go further to reach another endpoint or switch.
- System A sends an Exchange Type admin micropacket to the 3rd closest point with an element address of 0xFFFFFFFF and a hop-count of 3. This will be received and processed by System B. System B will respond in the Return Type admin micropacket that it is an endpoint. System A now knows where to exchange information regarding logical addresses.

In the above example, System A presumably would have been aware that the most directly attached component (Link-End A) is part of its own configuration and that it need not communicate with that component. It therefore would not have needed to start with a hop-count of 1 (but was not detrimentally affected by doing so).

System B could determine the type of device for System A in two ways:

- System B could duplicate the above steps in reverse.
- System B could use the information provided in the Exchange Type command that System A sent to System B when the third step in the above exchange took place. Endpoints should not wait for the other end to perform an exchange, but if the exchange occurs at an appropriate time, they may take advantage of the occurrence.

7.2 Logical address exchange

Once the other end of the link has been identified as to type (switch or non-switch endpoint), logical addresses are configured.

7.2.1 Endpoints on both ends

If both ends of the links are endpoints, each side shall use any desired value(s) for a Source ULA (excluding reserved address values). Although duplicate Source ULAs between the two ends of the link are possible, having the same address(es) at both ends will not prevent proper HIPPI-6400 operation.

If the other end of a link cannot be determined (the last reachable element does not respond properly as a switch or endpoint), the non-responsive end shall be treated as an endpoint.

7.2.2 Switches on both ends

If both ends are switches, address configuration is handled outside of this standard. Methods of switch configuration could include static manual table entry or automated address learning algorithms.

7.2.3 Endpoint to switch

If endpoints discover that they are connected to switches, they shall advertise a Source ULA. The ULA offer shall be made by sending a Return Logical Address admin micropacket.

Mechanisms for selection of this advertised ULA are not specified by this document and any desired approach may be used. A common approach of network equipment vendors is to use a ULA from a block of ULAs purchased from the IEEE. This method provides some guarantee of uniqueness and is recommended unless there are factors that require a different approach.

Upon receipt of a Return Logical Address admin micropacket, the receiver shall respond with a Logical Address Response admin micropacket. The Logical Address Response shall contain a Source ULA valid for the Logical Address Response recipient. This ULA may be the same as advertised in the original Return Logical Address offer or it may be different.

This returned Source ULA shall be accepted and subsequently used in all HIPPI-6400 messages by the receiver of the Logical Address Response admin micropacket.

Regardless of whether the returned Source ULA is the same as the Source ULA originally offered by the endpoint, the switch is the final selector of the address that will be used by the endpoint.

Annex A
(informative)

Switching

A.1 General

HIPPI-6400 switching of Messages is accomplished by processing the Destination ULA field of the HIPPI-6400-PH MAC header. This may be done based on the complete contents of the Destination ULA (48 bits) or on a subset of the field.

If a subset of the Destination ULA is used for switching, switches must ensure that Source ULAs are unique in the portion of the ULA operated on by the switch. Section 7.0 on page 7 describes the process of address configuration that gives switches final authority in configuration of Source ULAs.

When connections are made to other networks, the address range of the two (or more) networks is limited by the smaller of the connected address ranges.

For example, HIPPI-PH devices can be switched to communicate with HIPPI-6400 devices so long as all of the communicating devices restrict their addresses to 12 bits. The total number of devices is therefore limited to 4096 (minus reserved addresses).

The Destination ULA field in the Header micro-packet is used to control HIPPI-6400 physical layer switches, supporting the interconnection of many HIPPI-6400 devices. Figure 5 shows an example configuration that will be used to describe how HIPPI-6400 switches function. Three hosts and two switches are shown, actual configurations may be smaller or larger.

Although there is only a single mode of operation (logical addressing) specified for HIPPI-6400, users can achieve a form of source routing (as described in HIPPI-SC) by their selection of port configuration.

A.2 Logical addressing

Logical addresses specify where a Message is to be delivered, not the path to take to get there. Originating Sources use the same logical address to reach a Final Destination, no matter where the Originating Source is located.

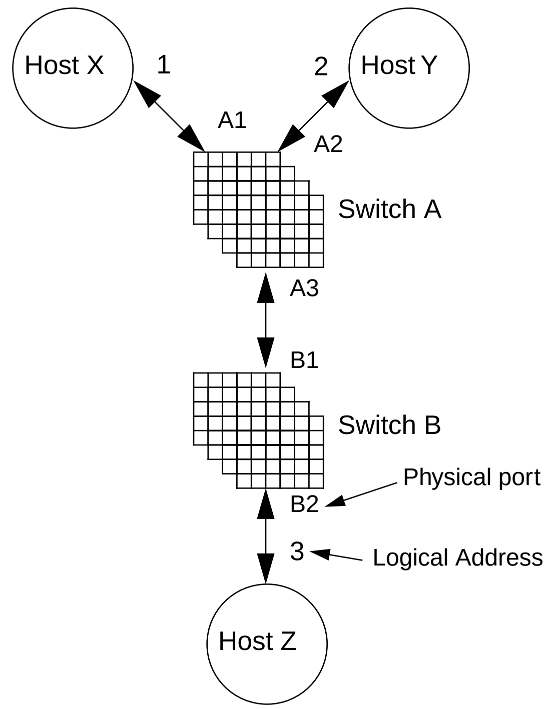

Figure 5. Hosts and switch configuration

In Figure 5, Host X, Host Y, and even Host Z can use logical address number “3” to specify that a Message should be sent to Host Z.

With logical addresses, the intermediate switches are responsible for selecting an appropriate path.

It is envisioned that switches can be built to use look-up tables at each input port to map addresses to Destinations. A look-up table can be indexed using the Destination ULA field. The look-up table would be used to hold a possible path(s) for a Destination.

A major advantage of using logical addresses is that only the switches need to know the fabric interconnection topology and the hosts only need to know the logical addresses. Hence if a link or port fails, switches can address around it without the hosts having to know about it or do anything special.

A.3 Input specific logical addressing

Because each input port is specified to contain a unique address look-up capability, it is possible to use logical switch addressing for limited source routing. Note that only the input portion of a port is involved in addressing. When a Message exits on a particular output port, it crosses that link without further addressing until received at the next input.

This capability means that it is possible to create addressing that could result in infinite looping of a micropacket. This will rarely be desirable and should be avoided.

One possible use of input port specific routing is to provide a test capability for monitoring the performance of specific links. In Figure 5, if Host Y wants to monitor the state of the link between switch A and switch B, he can send a Message to switch A and then to switch B. Port B1's address table (at switch B) can direct the Message back to B1, then switch A, and back to Host Y. To do this, the same address must be handled differently by individual ports. Table 6 shows a simplified look-up table that would work in this example.

Table 6. Port look-up table

Logical Address	Port Number	Destination
2	A2	A3
2	B1	B1
2	A3	A2

Because there are many available addresses, normal flat addressing can be used for host communications with other addresses used to support input specific logical routing for test and monitoring purposes.

Annex B
(informative)

Bridging

B.1 General

I believe that bridging with HIPPI-6400, as described here is no longer possible (the destination ULA is changed). Is this true? Is there something else that needs to go here?

HIPPI-6400 bridging may be used as a substitute for directly manipulating MAC addresses of connected media types. Bridges use ULAs embedded in the Message body as a look-up for the current media address.

With bridging, the incoming MAC address is only used to send across a single fabric. At each media translation to and/or from HIPPI-6400, a new MAC address is found based on the Message ULA address.

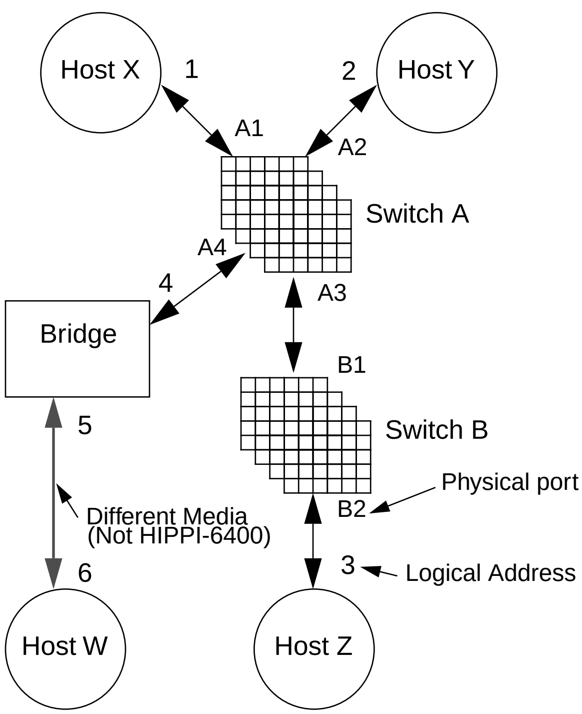

Figure 6. Hosts, switch, and bridge configuration

For example, in Figure 6, Host X uses a logical address of “4” to communicate with the bridge device when sending a message to Host W (logical address “6” on a separate network). The bridge operates on a ULA contained in the message body to look up the address of Host W on the connected network. Host W would reply using the bridge address (“5”) and the bridge would look up the logical address of X (“1) to place in the HIPPI-6400 MAC header

Building of look-up tables for bridging operations can be done using an automated process such as ARP or can be handled with static table entries.

Annex C
(informative)

Routing

C.1 General

Routing uses a logically assigned address (an example would be an IP address) for sending Messages. Routers keep a look-up of media ports with associated MAC addresses for given logical addresses.

With routing, the incoming MAC address is only used to send across a single fabric. At each media translation to and/or from HIPPI-6400, a new MAC address is found based on the logical address.

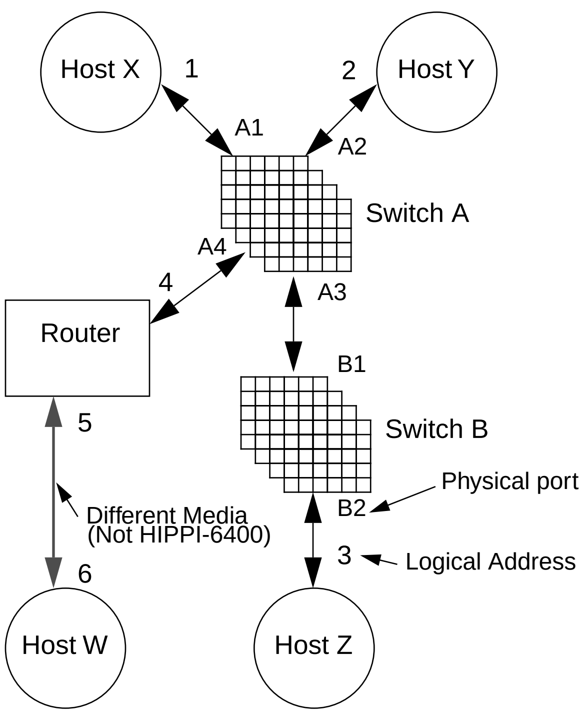

Figure 7. Hosts, switch, and router configuration

The example shown in Figure 7 is very similar to the previous example discussed in bridging. However, with routing, the operations are carried out on logical addresses (like IP).

Building of look-up tables for routing operations can be done using an automated process such as RIP or can be handled with static table entries.